Discrete Applied Mathematics 25 (1989) 321-323
North-Holland

NOTE

AN $O(n^{1.5})$ ALGORITHM TO COLOR PROPER CIRCULAR ARCS*

Wei-Kuan SHIH

Institute of Information Science, Academia Sinica, Republic of China

Wen-Lian HSU

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60201, USA

Received 14 June 1988
Revised 13 December 1988

A circular-arc family F is a collection of arcs on a circle. Such a family is said to be proper if no arc is contained within another. The coloring problem on circular-arc families has been shown to be NP-hard by Garey, Johnson, Miller and Papadimitriou. However, on proper circular-arc families, the coloring problem was shown to be solvable originally in $O(n^2 \log n)$ time by Orlin et al. and later, in $O(n^{1.5} \log n)$ time by Teng and Tucker. Their approach is to transform the (q-colorability) problem of testing whether G is q-colorable into a shortest path problem and, then, to apply binary search to solve $\log n$ q-colorability problems. In this note we show that Teng and Tucker's algorithm can be implemented to run in $O(n^{1.5})$ time overall using an interesting relationship between the time it takes to solve a q-colorability problem and the range of possible q to be searched for.

A circular-arc family F is a collection of arcs on a circle. An arc family is said to be *proper* if no arc is contained within another. A graph G is a (proper) *circular-arc graph* if there is a (proper) circular-arc family F and a one-to-one mapping of the vertices of G and the arcs in F such that two vertices in G are adjacent if and only if their corresponding arcs in F overlap. Circular-arc graphs have applications in compiler design and traffic light sequencing. Various characterization and optimization problems on circular-arc graphs have been studied [2]. The coloring problem on circular-arc graphs has been shown to be NP-hard by Garey et al. [1]. However, on proper circular-arc graphs, this problem was shown to be solvable in $O(n^2 \log n)$ time by Orlin et al. [3] given the arc representation. Their approach is to transform the (*q-colorability*) problem of testing whether G is q-colorable into a shortest path problem, which can be solved in $O(n^2)$ time and then to apply binary search to solve only $\log n$ q-colorability problems. Later, Teng and Tucker

* This research was supported in part by the National Science Foundation under Grant DMS-8504691 and by the National Science Council of the Republic of China while the second author visited the Academia Sinica.

[4] improved their q-colorability algorithm to run in $O(n^{1.5})$ time and obtained an $O(n^{1.5}\log n)$-coloring algorithm. In this note, we show that Teng and Tucker's algorithm can be implemented to run in $O(n^{1.5})$ time overall using an interesting relationship between the time it takes to solve a q-colorability problem and the range of possible q to be searched for.

Throughout this section we assume that G is a proper circular-arc graph with n vertices. Assume that the endpoints of all arcs are distinct and sorted in the clockwise direction starting with a chosen initial arc. Each arc has two endpoints. Define the *counterclockwise endpoints* of an arc a_i to be the first endpoint encountered in counterclockwise traversal from any interior point of a_i. For each arc a_i, define the set S_i of arcs (including a_i), each of which contains the counterclockwise endpoint of a_i, to be the *overlapping set* associated with a_i. Let s be $\max_{1\le i\le n}\{|S_i|\}$ and $\gamma(G)$ be the minimum number of colors needed to color the graph G. We shall use the following two results of [4,5]. The reader is referred to these two papers for detailed proofs.

Theorem 1 (Teng and Tucker [4]). *For any integer $q<n$, the q-colorability problem can be solved in* $\min\{q, n/q\}\cdot n$ *time.*

In Teng and Tucker's algorithm [4], $\lfloor\sqrt{n}\rfloor$ is a threshold for the quantity q and each q-colorability problem can be solved in at most $O(n^{1.5})$ time. Furthermore, when q is not close to $\lfloor\sqrt{n}\rfloor$, the q-colorability problem can be solved more efficiently. This observation together with the range estimates of the next theorem constitute the basis for our approach.

Theorem 2 (Tucker [5]). $\gamma(G)\le s+\lceil(s-1)/k\rceil$, *where* $k=\lfloor n/s\rfloor$. *Furthermore, if* $n\ge s^2$, *then* $\gamma(G)\le s+1$.

Tucker [5] showed that, if s divides n, then the greedy algorithm can optimally color G using s colors. If n is not a multiple of s (let the residue r be $n-s\cdot\lfloor n/s\rfloor=n-sk$), then deleting r arcs from G allows one to color the remaining sk arcs using s colors. One way to select these r arcs is to take k independent arcs (and assign one new color to them) at a time. These would guarantee that G can be colored by $s+\lceil r/k\rceil\le s+\lceil(s-1)/k\rceil$ colors. We shall make use of this theorem to reduce the range of possible q.

Lemma 3. *Assume* $s\in[a,b]$ *and* $a\ge\lceil\sqrt{n}\rceil$. *Then* $\gamma(G)\in[s, s+(b^2/(n-b)+1)]$ *and it takes at most* $O(n^2/a)$ *time to test if G is q-colorable for any* $q\ge s$.

Proof. Because $q\ge s\ge a\ge\lceil\sqrt{n}\rceil$, by Theorem 1, the running time of the q-colorability algorithm by Teng and Tucker [4] is at most $O(n^2/q)$, which is bounded by $O(n^2/a)$. The quantity $\lceil(s-1)/k\rceil$ is bounded by $s^2/(n-s)+1$. Now, it is easy to verify that, since $b\ge s$, we have $s^2/(n-s)\le b^2/(n-b)$. □

We shall make use of Lemma 3 to determine some strategical a_i and b_i such that the union of intervals $[a_1, b_1]$, $[a_2, b_2], \ldots, [a_k, b_k]$ is the interval $[0, n]$, where k is a function of n and that, if s falls within any $[a_i, b_i]$, the total running time can be bounded by $O(n^{1.5})$ using Lemma 3. We need the following notations. Let $\log^{(1)} n$ be $\log n$ and for $i > 1$, let $\log^{(i)} n$ be $\log(\log^{(i-1)} n)$. Thus, $\log^{(i)} n$ is the result of taking the log function iteratively i times on n. Define $\log^* n$ to be the least integer i such that $\log^{(i)} n \le 1$. The intervals for s are divided as follows:

(1) $\lceil \sqrt{n} \cdot \log n \rceil \le s \le n$. Then for each $q \ge s$, the q-colorability problem can be solved in $O(n^{1.5}/\log n)$ time. Since we need to test at most $\log n$ q-colorability problems, $\gamma(G)$ can be found in $O(n^{1.5})$ time.

(2) $0 \le s \le \lceil \sqrt{n} \rceil$. Then, by Theorem 2, $\gamma(G) \le s + 1$ and we only need to test two q-colorability problems for $q = s, s+1$. Hence, $\gamma(G)$ can be found in $O(n^{1.5})$ time.

(3) $\lceil \sqrt{n} \rceil \le s \le \lceil \sqrt{n} \cdot \log n \rceil$. Denote $\log^* n$ by d. Then s must be in one of the following $d-1$ intervals: $[\lceil \sqrt{n} \cdot \log^{(d)} n \rceil, \lceil \sqrt{n} \cdot \log^{(d-i)} n \rceil]$, $[\lceil \sqrt{n} \cdot \log^{(d-1)} n \rceil, \lceil \sqrt{n} \cdot \log^{(d-2)} n \rceil], \ldots, [\lceil \sqrt{n} \cdot \log^{(2)} n \rceil, \lceil \sqrt{n} \cdot \log n \rceil]$. We use the next corollary to determine the search range for each interval $[\lceil \sqrt{n} \cdot \log^{(i+1)} n \rceil, \lceil \sqrt{n} \cdot \log^{(i)} n \rceil]$.

Corollary 4. *If* $s \in [\lceil \sqrt{n} \cdot \log^{(i+1)} n \rceil, \lceil \sqrt{n} \cdot \log^{(i)} n \rceil]$, $1 \le i < d$, *then* $\gamma(G) \in [s, s + 2(\log^{(i)} n)^2]$.

Proof. By Lemma 3, the search range for $\gamma(G)$ is

$$\le \frac{(\lceil \sqrt{n} \cdot \log^{(i)} n \rceil)^2}{n - \lceil \sqrt{n} \cdot \log^{(i)} n \rceil} + 1 \le \frac{(\sqrt{n} \cdot \log^{(i)} n + 1)^2}{n - \sqrt{n} \cdot \log^{(i)} n} + 1 \le 2(\log^{(i)} n)^2. \quad \square$$

Within each interval, the search range is $2(\log^{(i)} n)^2$ for $i < d$. Hence the number of q-colorability problems to be solved (based on binary search) is bounded by $2 \log^{(i+1)} n + 1$. The time it takes to solve each q-colorability problem is bounded by $O(n^2/(\sqrt{n} \cdot \log^{(i+1)} n)) = O(n^{1.5}/\log^{(i+1)} n)$. Hence, the total running time is bounded by $O(n^{1.5})$.

References

[1] M. Garey, D. Johnson, G. Miller and C. Papadimitriou, The complexity of coloring circular arcs and chords, SIAM J. Algebraic Discrete Methods 2 (1980) 216–227.

[2] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs (Academic Press, New York, 1980).

[3] J. Orlin, M. Bonuccelli and D. Bovel, An $O(n^2)$ algorithm for coloring proper circular arc graphs, SIAM J. Algebraic Discrete Methods 2 (1981) 88–93.

[4] A. Teng and A. Tucker, An $O(qn)$ algorithm to q-color a proper family of circular arcs, Discrete Math. 55 (1985) 233–243.

[5] A. Tucker, Coloring a family of circular-arcs, SIAM J. Appl. Math. 29 (1975) 1–24.